お客様各位

資料中の「ラピスセミコンダクタ」等名称の ラピステクノロジー株式会社への変更

2020 年 10 月 1 日をもって、ラピスセミコンダクタ株式会社の LSI 事業部門は、ラピステクノロジー株式会社に分割承継されました。従いまして、本資料中にあります「ラピスセミコンダクタ株式会社」、「ラピスセミ」、「ラピス」といった表記に関しましては、全て「ラピステクノロジー株式会社」に読み替えて適用するものとさせていただきます。なお、会社名、会社商標、ロゴ等以外の製品に関する内容については、変更はありません。以上、ご理解の程よろしくお願いいたします。

2020年10月1日
ラピステクノロジー株式会社

Dear customer

LAPIS Semiconductor Co., Ltd. ("LAPIS Semiconductor"), on the 1st day of October, 2020, implemented the incorporation-type company split (shinsetsu-bunkatsu) in which LAPIS established a new company, LAPIS Technology Co., Ltd. (“LAPIS Technology”) and LAPIS Technology succeeded LAPIS Semiconductor’s LSI business.

Therefore, all references to "LAPIS Semiconductor Co., Ltd.", "LAPIS Semiconductor" and/or "LAPIS" in this document shall be replaced with "LAPIS Technology Co., Ltd."

Furthermore, there are no changes to the documents relating to our products other than the company name, the company trademark, logo, etc.

Thank you for your understanding.

LAPIS Technology Co., Ltd.
October 1, 2020

ROHM GROUP
LAPIS
SEMICONDUCTOR

FJXL_MCU_STARTUP-02

ML62Q1000 シリーズ
スタートアップファイルの解説

発行日 2020 年 3 月 19 日

ROHM
SEMICONDUCTOR

ご注意

1) 本資料の記載内容は改良などのため予告なく変更することがあります。

2) ラピスセミコンダクタは常に品質・信頼性の向上に取り組んでおりますが，半導体製品は種々の要因で故障・誤作動する可能性があります。
万が一，本製品が故障・誤作動した場合であっても，その影響により人身事故，火災損害等が起こらないようご使用機器でのディレーティング，冗長設計，延焼防止，バックアップ，フェイルセーフ等の安全確保をお願いします。定格を超えたご使用や使用上の注意書が守られていない場合，いかなる責任もラピスセミコンダクタは負うものではありません。

3) 本資料に記載されております応用回路例やその定数などの情報につきましては，本製品の標準的な動作や使い方を説明するものです。したがいまして，量産設計をされる場合には，外部諸条件を考慮していただきますようお願いいたします。

4) 本資料に記載されております技術情報は，本製品の代表的動作および応用回路例などを示したものであり，それをもって，当該技術情報に関するラピスセミコンダクタまたは第三者の知的財産権その他の権利を許諾するものではありません。したがいまして，上記技術情報の使用に起因して第三者の権利にかかわる紛争が発生した場合，ラピスセミコンダクタはその責任を負うものではありません。

5) 本製品は，一般的な電子機器（AV機器，OA機器，通信機器，家電製品，アミューズメント機器など）および本資料に明示した用途への使用を意図しています。

6) 本資料に掲載されております製品は，耐放射線設計はなされておりません。

7) 本製品を下記のような特に高い信頼性が要求される機器等に使用される際には，ラピスセミコンダクタへ必ずご連絡の上，承諾を得てください。
・輸送機器（車載，船舶，鉄道など），幹線用通信機器，交通信号機器，防災・防犯装置，安全確保のための装置，医療機器，サーバー，太陽電池，送電システム

8) 本製品を極めて高い信頼性を要求される下記のような機器等には，使用しないでください。
・航空宇宙機器，原子力制御機器，海底中継機器

9) 本資料の記載に従わないために生じたいかなる事故，損害もラピスセミコンダクタはその責任を負うものではありません。

10) 本資料に記載されております情報は，正確を期すため慎重に作成したものですが，万が一，当該情報の誤り・誤植に起因する損害がお客様に生じた場合においても，ラピスセミコンダクタはその責任を負うものではありません。

11) 本製品のご使用に際しては，RoHS 指令など適用される環境関連法令を遵守の上ご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して，ラピスセミコンダクタは一切の責任を負いません。本製品の RoHS 適合性などの詳細につきましては，セールス・オフィスまでお問合せください。

12) 本製品および本資料に記載の技術を輸出または国外へ提供する際には，「外国為替および外国貿易法」，「米国輸出管理規則」など適用される輸出関連法令を遵守し，それらの定めにしたがって必要な手続を行ってください。

13) 本資料の一部または全部をラピスセミコンダクタの許可なく，転載・複写することを堅くお断りします。

ラピスセミコンダクタ株式会社
〒222-8575 神奈川県横浜市港北区新横浜 2-4-8
http://www.lapis-semi.com

目次

1. はじめに

本ドキュメントは，ラピスセミコンダクタが提供する各商品のスタートアップファイルをカスタマイズする際の参考になるよう，スタートアップファイルの内容について説明しています。

2. 対象商品

ML62Q1000 シリーズ
- -ML62Q1200E グループ
- -ML62Q1200A グループ
- -ML62Q1300 グループ
- -ML62Q1400 グループ
- -ML62Q1500/ML62Q1800 グループ
- -ML62Q1600 グループ
- -ML62Q1700 グループ

3. 関連マニュアル類

ML62Q1000 シリーズ ユーザーズマニュアル

U8/U16 Development Tools
- -CCU8 プログラミングガイド
- -MACU8 アセンブラパッケージ ユーザーズマニュアル

アプリケーションノート
- -ソフトウェア安全設計

LAPIS Code Generation Tools
- -ML62Q1000 スタートアップ設定ツール ユーザーズマニュアル

4. スタートアップファイルについて

4.1 解説

ここでは, ML62Q1367 のスタートアップファイルを実例として取り上げ, スタートアップファイル中の各行が具体的に何をしているのかを解説します。
スタートアップファイルの詳細については, CCU8 プログラミングガイドの「2.5 スタートアップの説明」を参照してください。

【語句参照】
(*1) 疑似命令 : 詳細については, MACU8 アセンブラパッケージ ユーザーズマニュアルの「5 疑似命令の詳細」を参照してください。
(*2) リロケータブルセグメント : 詳細については, MACU8 アセンブラパッケージ ユーザーズマニュアルの「2.4 論理セグメント」を参照してください。

```
;*****************************************************************
;
;           ML621367 start up assembly source file
;           for CCU8 version 1.xx/2.xx/3.xx
;           SMALL CODE MODEL
;           ROMWINDOW    00000H-0DFFFH
;           Version 1.00
;           Last Edition    Oct 30, 2018
;           Copyright (C) 2018 LAPIS Semiconductor Co., Ltd.
;           U8/U16 Device Information Files V2.00
;*****************************************************************
;
            type(ML621367)

            model     small
            romwindow           0, 0dfffh

            extrn     code: _main
            extrn     data near: _$$SP

            public    $$start_up


            cseg      at 0:0h
            dw        offset _$$SP

            cseg      at 0:4h
            dw        $$brk_reset

            cseg      at 0:6h
            dw        0ffffh


$$NCODml621367sw segment code #0
            rseg      $$NCODml621367sw

$$start_up:
            tb        INITE
            bz        $begin

$$brk_reset:
$$error:
            mov       r0,       #00h
            st        r0,       WDTMOD
            mov       psw,      #02h
```

TYPE 疑似命令(*1)で対象商品を指定します。

MODEL 疑似命令でメモリモデルを指定します。
ROMWINDOW 疑似命令で ROM ウィンドウ領域の範囲を指定します。
_main を外部参照することを宣言します。
_$$SP（スタックポインタの初期値）を外部参照することを宣言します。
$$start_up（スタートアップのラベル）が別ファイルから参照されることを宣言します。

0000H 番地にスタックポインタの初期値（_$$SP）を定義します。

0004H 番地（BRK 命令のベクタアドレス）に $$brk_reset のアドレスを定義します。

0006H 番地（nmice のベクタアドレス）に FFFFH（BRK 命令）を定義します。

リロケータブルセグメント(*2)（$$NCODml621367sw）を定義します。

マイコンの起動に異状があったか確認します。
INITE=”1 の場合，WDT のリセットを待ちます。

```
    ;---------------------------------------------------
    ;       reset SFRs
    ;       If you don't need the below then remove it.
    ;
            mov     er0,    #-1
            st      er0,    BRECON0
            st      er0,    BRECON1
            st      er0,    BRECON2
            st      er0,    BRECON3
    ;---------------------------------------------------
            bal     $                               ; wait for Watch Dog Timer reset

$begin:

;---------------------------------------------------------------
;           setting Memory Model
;---------------------------------------------------------------
; nothing (fixed as Small model)

;---------------------------------------------------------------
;           setting Rom Window range
;---------------------------------------------------------------
; nothing (fixed as range 0-0dfffh)

;---------------------------------------------------------------
;           user SFR definition
;---------------------------------------------------------------
;
; In C source file, define this function as follows:
;
; void setting_SFR(void)
; {
;       ...
; }
;---------------------------------------------------------------
; extrn code : _setting_SFR
;       bl      _setting_SFR

;---------------------------------------------------------------
;           Data memory zero clear
;---------------------------------------------------------------
NEAR_RAM_START  equ     0e000h
NEAR_RAM_END    equ     0efffh

            mov     er0,    #0
            mov     er2,    #0
            mov     er4,    #0
            mov     er6,    #0

            mov     r8,     #BYTE1 NEAR_RAM_START
            mov     r9,     #BYTE2 NEAR_RAM_START
            lea     [er8]
__near_ram_loop:
            st      qr0,    [ea+]
            add     er8,    #8                  ;er8 += 8
            cmp     r9,     #BYTE2 (NEAR_RAM_END+1)
            blt     __near_ram_loop
            cmp     r8,     #BYTE1 (NEAR_RAM_END+1)
            blt     __near_ram_loop

;---------------------------------------------------------------
;           WDT counter clear
;
```

RAM 領域を”0”クリアします。
マイコンに搭載している RAM 容量に依存して処理時間が変わります。
本ドキュメントの「4.2 実行時間」を参照してください。

```
;           !!NOTICE!!
;           After this process, you have to clear WDT counter within
;           the time described in user's manual of LSI.
;
;           !!NOTICE!!
;           Please remove this routine when you use DTU8
simulation
;           mode.
;-------------------------------------------------------------
;__wait_wdt_clear:
;            tb          WDTCLR1
;            bnz         __wait_wdt_clear
;            mov         r0,         psw
;            di
;            mov         r12,        #05ah
;            mov         r13,        #0a5h
;            st          r12,        WDTCON
;            st          r13,        WDTCON
;            mov         psw,        r0

;-------------------------------------------------------------
;           data variable initialization
;-------------------------------------------------------------
            mov         r10,      #SEG $$init_info
            lea         OFFSET $$init_info
__init_loop:
            ;-------------------------------------------
            ; get source offset address and set in ER0
            ;-------------------------------------------
            l           er0,      r10:[ea+]
            cmp         r0,       #0ffh
            bne         __skip
            cmp         r1,       #0ffh
            beq         __init_end                    ;if er0==0ffffh then
exit
__skip:
            ;-------------------------------------------
            ; get destination offset address and set in ER2
            ;-------------------------------------------
            l           er2,      r10:[ea+]

            ;-------------------------------------------
            ; get size of objects and set in ER4
            ;-------------------------------------------
            l           er4,      r10:[ea+]

            ;-------------------------------------------
            ; get source phy_seg address and set in R6
            ;-------------------------------------------
            l           r6,       r10:[ea+]

            ;-------------------------------------------
            ; get destination phy_seg address and set in R7
            ;-------------------------------------------
            l           r7,       r10:[ea+]

            ;-------------------------------------------
            ; copy
            ;-------------------------------------------
            add         er4,      #0
            bz          __init_loop                   ;if er4==0000 then
next
```

ソフトウェア安全設計より，メインループ内の一箇所で WDT をクリアすることを推奨しています。よって，スタートアップファイル内での WDT クリアはマスクしています。
詳細については，ソフトウェア安全設計の「3.1 ウォッチドッグタイマ（WDT）のクリア処理」を参照してください。

初期値付き変数を初期化します。

```
        tb      r2.0
        bnz     __loop_by_byte
        tb      r4.0
        bnz     __loop_by_byte

__init_loop2:
        l       er8,    r6:[er0]
        add     er0,    #2              ;er0 += 2
        st      er8,    r7:[er2]
        add     er2,    #2              ;er2 += 2

        add     er4,    #-2             ;er4 -= 2
        bnz     __init_loop2
        bal     __init_loop

__loop_by_byte:
        l       r8,     r6:[er0]
        add     er0,    #1              ;er0 += 1
        st      r8,     r7:[er2]
        add     er2,    #1              ;er2 += 1

        add     er4,    #-1             ;er4 -= 1
        bnz     __loop_by_byte
        bal     __init_loop

__init_end:

;------------------------------------------------------------
;       call initializing routine
;------------------------------------------------------------
        bl      $$content_of_init

;------------------------------------------------------------
;       initialize DSR zero
;------------------------------------------------------------
        mov     r0,     #0
        st      r0,     DSR

;------------------------------------------------------------
;       far jump to main routine
;------------------------------------------------------------
        b       _main

;------------------------------------------------------------
;       segment definition for initializing routine
;------------------------------------------------------------
$$content_of_init segment code
        rseg    $$content_of_init

$$end_of_init segment code
        rseg    $$end_of_init
        rt

;------------------------------------------------------------
;       segment definition for data variable initialization
;------------------------------------------------------------
$$init_info segment table 2
        rseg    $$init_info
        dw      $$NINITTAB
        dw      $$NINITVAR
        dw      size $$NINITTAB
        db      seg $$NINITTAB
```

C ソースプログラムの ABSOLUTE プラグマや SEGINIT プラグマにより，絶対番地に割り付けられた初期値付き変数を初期化します。

C ソースプログラムの MAIN へ移行します。

$$content_of_init の内容（初期値付き変数に初期値を代入する処理）はコンパイラにより生成されます。

$$end_of_init は，リンカにより$$content_of_init の最後にリンクされます。

C 言語の初期値付き変数の初期値を格納するためのテーブルを定義しています。詳細については，CCU8 プログラミングガイドの「2.5.14 セグメントの定義」を参照してください。

```
        db      seg $$NINITVAR

$$init_info_end segment table
        rseg    $$init_info_end
        dw      0ffffh

$$NINITVAR segment data 2 #0

$$NINITTAB segment table 2


;-------------------------------------------------------------
;       Setting the code-option data (for ML621367)
;-------------------------------------------------------------
        cseg #0 at 0ffc0h       ; address
        dw      0ffffh           ; 0ffc0h
        dw      0ffffh           ; 0ffc2h
        dw      0ffffh           ; 0ffc4h
        dw      0ffffh           ; 0ffc6h
        dw      0ffffh           ; 0ffc8h
        dw      0ffffh           ; 0ffcah
        dw      0ffffh           ; 0ffcch
        dw      0ffffh           ; 0ffceh

        cseg #0 at 0ffd0h       ; address
        dw      0fffdh           ; 0ffd0h

        cseg #0 at 0ffd2h       ; address
        dw      0ffffh           ; 0ffd2h
        dw      0ffffh           ; 0ffd4h
        dw      0ffffh           ; 0ffd6h
        dw      0ffffh           ; 0ffd8h
        dw      0ffffh           ; 0ffdah
        dw      0ffffh           ; 0ffdch
        dw      0ffffh           ; 0ffdeh

        cseg #0 at 0ffe0h       ; address
        dw      0ffffh           ; 0ffe0h
        dw      0ffffh           ; 0ffe2h
        dw      0ffffh           ; 0ffe4h
        dw      0ffffh           ; 0ffe6h
        dw      0ffffh           ; 0ffe8h
        dw      0ffffh           ; 0ffeah
        dw      0ffffh           ; 0ffech
        dw      0ffffh           ; 0ffeeh
        dw      0ffffh           ; 0fff0h
        dw      0ffffh           ; 0fff2h
        dw      0ffffh           ; 0fff4h
        dw      0ffffh           ; 0fff6h
        dw      0ffffh           ; 0fff8h
        dw      0ffffh           ; 0fffah
        dw      0ffffh           ; 0fffch
        dw      0ffffh           ; 0fffeh

;-------------------------------------------------------------
;       Keeping the Mirror area (for ML621367)
;-------------------------------------------------------------
        tseg #08h at 00000h
        ds      10000h

        tseg #09h at 00000h
        ds      00400h
```

$$init_info_end は，初期化情報テーブルの終了を示すもので，リンカにより初期化情報テーブル（$$init_info）の最後にリンクされます。

初期値付き変数の変数領域を示すセグメントを定義します。
初期値付き変数の初期化テーブル領域を示すセグメントを定義します。

コードオプションを設定します。
コードオプションの詳細については，
ML62Q1000 シリーズ ユーザーズマニュアルの「26. コードオプション」を参照してください。

セグメント 8 の 00000H 番地～0FFFFH 番地（ミラー領域）にコードが配置されないように領域を確保しています。
セグメント 9 の 00000H 番地～003FFH 番地（テ

スト領域のミラー領域）にコードが配置されないように領域を確保しています。

```
;-------------------------------------------------------------
;           Keeping the Self-Programming area (for ML621367)
;-------------------------------------------------------------
            tseg #1fh at 00000h
            ds          00800h
```

セグメント 31 の 00000H 番地～007FFH 番地（データ・フラッシュ領域）にコードが配置されないように領域を確保しています。

```
;-------------------------------------------------------------
;           Keeping the Test area over #1 (for ML621367)
;-------------------------------------------------------------
            tseg #01h at 00000h
            ds          00400h
```

セグメント 1 の 00000H 番地～003FFH 番地（テスト領域）にコードが配置されないように領域を確保しています。

```
            end
```

END 疑似命令で終了します。

4.2 実行時間

表 1 に電源起動時からスタートアップの処理が完了するまでの時間を示します。電源起動時は低速 RC 発振クロックを 512 カウント（約 16ms）してから，CPU がプログラムを実行します。なお，低速 RC 発振クロックの周波数には個体差がありますので，表 1 の数値は目安となります。

スタートアップの処理時間は RAM 容量とシステムクロックに依存します。ML62Q1000 シリーズは低速クロック（LSCLK）で CPU がプログラム処理を開始します。スタートアップを短い時間で処理したい場合，RAM 領域を”0”クリアする前に，システムクロックを高速クロック（HSCLK）に切り替えてください。

システムクロックの切り替え方法は，ML62Q1000 シリーズ ユーザーズマニュアルの「6.3.4 システムクロックの切り替え」を参照してください。
あるいは，LAPIS Code Generation Tools を利用すれば，システムクロックを高速クロック（HSCLK）に切り替える処理を，容易にスタートアップファイルに追加することができます。

表 1　電源起動時からスタートアップの処理が完了するまでの時間（1/2）

CPU 動作モード：ウェイトモード

RAM サイズ [K バイト]	システムクロック	実行時間[ms]
2	32.768kHz	120
	16MHz	18.0
	24MHz	18.0
4	32.768kHz	222
	16MHz	18.1
	24MHz	18.1
8	32.768kHz	424
	16MHz	18.6
	24MHz	18.5
16	32.768kHz	830
	16MHz	19.6
	24MHz	19.1
32	32.768kHz	1630
	16MHz	21.4
	24MHz	20.2

表 1　電源起動時からスタートアップの処理が完了するまでの時間（2/2）

CPU 動作モード：ノーウェイトモード

RAM サイズ [K バイト]	システムクロック	実行時間[ms]
2	32.768kHz	80
	6MHz	18.0
	8MHz	18.0
4	32.768kHz	142
	6MHz	18.2
	8MHz	18.2
8	32.768kHz	268
	6MHz	18.8
	8MHz	18.5
16	32.768kHz	516
	6MHz	20.4
	8MHz	19.6
32	32.768kHz	1015
	6MHz	23.0
	8MHz	21.8

5. 改版履歴

ドキュメント No.	発行日	ページ		変更内容
		改版前	改版後	
FJXL_MCU_STARTUP-01	2019.3.11	ー	ー	初版発行
FJXL_MCU_STARTUP-02	2020.3.19	*	*	誤記訂正